

## CLAIMS

What is claimed is:

1. A method of forming an executable program from a plurality of object code modules, each object code module having section data, a set of relocation instructions, and one or more symbols, each symbol having a plurality of attributes associated therewith, wherein said relocation instructions include a data retrieval instruction having a symbol field identifying a symbol and an attribute field identifying a symbol attribute associated with said identified symbol to be retrieved, the method comprising:

reading at least one relocation instruction from said set of relocation instruction and where said relocation instruction is a data retrieval instruction, determining the symbol identified by the symbol field and retrieving one of said plurality of symbol attributes associated with said symbol in dependence on the contents of the symbol attributes field of said instruction.

2. The method of claim 1 wherein said retrieved symbol attribute is placed in a store for subsequent use by a further relocation instruction.

3. The method of claim 2 wherein said store is a stack.

4. The method of claim 1 wherein said method further comprises recording a pass value indicative of the number of times said set of relocation instructions from said plurality of object code modules have been read.

5. The method of claim 1 wherein said plurality of symbol attributes includes the value of the symbol.

6. The method of claim 1 wherein said plurality of symbol attributes includes the name of the symbol itself.

7. The method of claim 1 wherein said plurality of symbol attributes includes a ranking determinator, said ranking determinator defining which one of a plurality of definitions of said symbol is selected when forming said executable program.

8. The method of claim 4 wherein said plurality of symbol attributes includes said pass value indicative of the most recent repetition of said set of relocation instructions during which the value of said symbol has been retrieved.

9. The method of claim 8 wherein said method further comprises determining if the pass value of claim 8 is equal to or only one less than said recorded pass value and in response to said determination placing a predetermined value in said store.

10. The method of claim 9, further comprising reading said predetermined value placed in said store and deleting the section data labeled by said symbol in response to the value of said predetermined value.

11. A method of forming an executable program from a plurality of object code modules, each object code module comprising section data, a set of relocation instructions, and one or more symbols, each symbol having a plurality of symbol attributes associated therewith, said symbol attributes including said symbol value, wherein said relocation instructions include a data retrieval instruction having a symbol field identifying one of said symbols and an attribute field identifying one of said plurality of symbol attributes associated with said identified symbol to be retrieved, the method comprising:

reading at least one relocation instruction from said set of relocation instructions;

recording a pass value indicative of the number of times said set of relocation instructions have been read;

where said relocation instruction is a data retrieval instruction, identifying the symbol identified by said symbol field, determining if said associated symbol value has been retrieved by a

further data retrieval instruction during the current or previous repetition of said set of relocation instructions, and responsive to said determination placing a predetermined value in a store.

12. The method of claim 11, further comprising:

when said symbol attribute field of said data retrieval instruction identifies said symbol value, storing said pass value in a further one of said symbol attributes.

13. A computer program product for forming an executable program from a plurality of object code modules, said computer program product comprising: program code means having section data, a set of relocation instructions, and one or more symbols, each symbol having a plurality of attributes associates therewith, wherein said relocation instructions include a data retrieval instruction having a symbol field identifying a symbol and an attribute field identifying a symbol attribute associated with said identified symbol to be retrieved, said program code means arranged so that, when run on a computer, at least one relocation is read from the relocation instructions, and where the relocation instruction is a data retrieval instruction, determining the symbol identified by the symbol field is determined and one of the plurality of symbol attributes associated with the symbol in dependence on the contents of the symbol attributes field of the instruction is retrieved.

14. A computer program product for forming an executable program from a plurality of object code modules, said computer program product comprising: program code means having section data, a set of relocation instructions, and one or more symbols, each symbol having a plurality of attributes associates therewith, wherein said relocation instructions include a data retrieval instruction having a symbol field identifying a symbol and an attribute field identifying a symbol attribute associated with said identified symbol to be retrieved, said program code means arranged so that, when run on a computer, at least one relocation is read from the relocation instructions, and where the relocation instruction is a data retrieval instruction, determining the symbol identified by the symbol field is determined and one of the plurality of symbol attributes associated with the symbol in dependence on the contents of the symbol attributes field of the

instruction, the retrieved symbol attribute being placed in a store for subsequent use by a further relocation instruction is retrieved.

15. A computer program product for forming an executable program from a plurality of object code modules, said computer program product comprising: program code means having section data, a set of relocation instructions, and one or more symbols, each symbol having a plurality of attributes associates therewith, wherein said relocation instructions include a data retrieval instruction having a symbol field identifying a symbol and an attribute field identifying a symbol attribute associated with said identified symbol to be retrieved, said program code means arranged so that, when run on a computer, at least one relocation is read from the relocation instructions, and where the relocation instruction is a data retrieval instruction, determining the symbol identified by the symbol field is determined and one of the plurality of symbol attributes associated with the symbol in dependence on the contents of the symbol attributes field of the instruction, the retrieved symbol attribute being placed in a stack for subsequent use by a further relocation instruction is retrieved.

16. A computer program product for forming an executable program from a plurality of object code modules, said computer program product comprising: program code means having section data, a set of relocation instructions, and one or more symbols, each symbol having a plurality of attributes associates therewith, wherein said relocation instructions include a data retrieval instruction having a symbol field identifying a symbol and an attribute field identifying a symbol attribute associated with said identified symbol to be retrieved, said program code means arranged so that, when run on a computer, at least one relocation is read from the relocation instructions, and where the relocation instruction is a data retrieval instruction, determining the symbol identified by the symbol field is determined and one of the plurality of symbol attributes associated with the symbol in dependence on the contents of the symbol attributes field of the instruction, the retrieved symbol attribute being placed in a stack for subsequent use by a further relocation instruction, and a pass value indicative of the number of times the relocation instructions from the plurality of object code modules have been read is recorded.

17. A computer program product for forming an executable program from a plurality of object code modules, said computer program product comprising: program code means having section data, a set of relocation instructions, and one or more symbols, each symbol having a plurality of attributes associates therewith, wherein said relocation instructions include a data retrieval instruction having a symbol field identifying a symbol and an attribute field identifying a symbol attribute associated with said identified symbol to be retrieved, said program code means arranged so that, when run on a computer, at least one relocation is read from the relocation instructions, and where the relocation instruction is a data retrieval instruction, determining the symbol identified by the symbol field is determined and one of the plurality of symbol attributes associated with the symbol in dependence on the contents of the symbol attributes field of the instruction, the retrieved symbol attribute being placed in a stack for subsequent use by a further relocation instruction, and a pass value indicative of the number of times the relocation instructions from the plurality of object code modules have been read is recorded, the plurality of symbol attributes including a ranking determinator, the ranking determinator defining which one of a plurality of definitions of the symbols is selected when forming an executable program.

18. A computer program product for forming an executable program from a plurality of object code modules, said computer program product comprising: program code means having section data, a set of relocation instructions, and one or more symbols, each symbol having a plurality of attributes associates therewith, wherein said relocation instructions include a data retrieval instruction having a symbol field identifying a symbol and an attribute field identifying a symbol attribute associated with said identified symbol to be retrieved, said program code means arranged so that, when run on a computer, at least one relocation is read from the relocation instructions, and where the relocation instruction is a data retrieval instruction, determining the symbol identified by the symbol field is determined and one of the plurality of symbol attributes associated with the symbol in dependence on the contents of the symbol attributes field of the instruction, the retrieved symbol attribute being placed in a stack for subsequent use by a further relocation instruction, and a pass value indicative of the number of times the relocation instructions from the plurality of object code modules have been read is recorded, the plurality of symbol

attributes including a ranking determinator, the ranking determinator defining which one of a plurality of definitions of the symbols is selected when forming an executable program, the plurality of symbol attributes also including a pass value indicative of the most recent repetition of the set of relocation instructions during which the value of the symbol has been retrieved, and further including a determination of whether the pass value is equal to or only one less than the recorded pass value, and in response to the determination placing the predetermined value in the store, reading the predetermined value placed in the store, and deleting the section data labeled by the symbol in response to the value of the predetermined value.

19. A computer program product for forming an executable program from a plurality of object code modules, said computer program product comprising program code means having section data, a set of relocation instructions and one or more symbols, each symbol having a plurality of symbol attributes associated therewith, said symbol attributes including said symbol value, wherein said relocation instructions includes a data retrieval instruction having a symbol field identifying one of said symbols and an attribute field identifying one of said plurality of symbol attributes associates with said identified symbol to be retrieved, said program code means being arranged so that, when run on a computer, it reads at least one relocation instruction from the set of relocation instructions; records a pass value indicative of a number of times the set of relocation instructions have been read; and with the relocation instruction as a data retrieval instruction, it identifies the symbol identified by the symbol field, determines if the associated symbol values has been retrieved by a further data retrieval instruction during the current or previous repetition of the set of relocation instructions, and places a predetermined value in a store in response to the determination.

20. A computer program product for forming an executable program from a plurality of object code modules, said computer program product comprising program code means having section data, a set of relocation instructions and one or more symbols, each symbol having a plurality of symbol attributes associated therewith, said symbol attributes including said symbol value, wherein said relocation instructions includes a data retrieval instruction having a symbol field

identifying one of said symbols and an attribute field identifying one of said plurality of symbol attributes associates with said identified symbol to be retrieved, said program code means being arranged so that, when run on a computer, it reads at least one relocation instruction from the set of relocation instructions; records a pass value indicative of a number of times the set of relocation instructions have been read; and with the relocation instruction as a data retrieval instruction, it identifies the symbol identified by the symbol field, determining if the associated symbol values has been retrieved by a further data retrieval instruction during the current or previous repetition of the set of relocation instructions, and placing a predetermined value in a store in response to the determination, and when the symbol attribute field of the data retrieval instruction identifies the symbol value, it stores the pass value in a further one of the symbol attributes.